Lab 3 (Bayes Classifiers & Boosting)

DD2421 Machine Learning

BAYESIAN LEARNING AND BOOSTING

- Naive Bayes:Assumes that features are conditionally independent given the class, estimates class-conditional probabilities and priors, and classifies samples by maximizing the posterior probability.
- AdaBoost: Iteratively trains weak learners slightly better than random, increases the weights of misclassified samples, and combines all learners through weighted voting to build a strong overall classifier.

Naive Bayes Classifier

A generative model assuming each class follows a Gaussian distribution:

$$p(x \mid y = k) = \mathcal{N}(x, \mid \mu_k \Sigma_k)$$

"Naive" assumption: feature independence → diagonal covariance

$$\Sigma_k = \mathrm{diag}\left(\sigma_{k1}^2, \ldots, \sigma_{kd}^2\right)$$

Parameter Estimation (Maximum Likelihood)

$$\pi_k = \frac{N_k}{N}, \quad \mu_k = \frac{1}{N_k} \sum_{i:y_i=k} x_i, \quad \sigma_{km}^2 = \frac{1}{N_k} \sum_{i:y_i=k} (x_{im} - \mu_{km})^2$$

Discriminant Function

$$\delta_k(x) = -\tfrac{1}{2} \sum_m \left[\log \sigma_{km}^2 + \frac{(x_m - \mu_{km})^2}{\sigma_{km}^2} \right] + \log \pi_k$$

Prediction rule: $\hat{y} = max_k \delta_k(x)$

AdaBoost Algorithm

- Initialize sample weights: $\omega_i = 1/N$
- Train weak classifier $h_t(x)$
- Compute weighted error:

$$\varepsilon_t = \sum_i \omega_i \; 1[h_t(x_i) \neq y_i]$$

- Compute learner weight:

$$\alpha_t = \frac{1}{2} \ln \frac{1 - \varepsilon_t}{\varepsilon_t}$$

- Update and normalize sample weights.
- Final prediction (multi-class version):

$$H(x) = \arg\max_k \sum_{t=1}^{T} \alpha_t \; 1[h_t(x) = k]$$

Assignment 1 & 2 - Parameter Estimation

- Estimate the mean μ and the diagonal variance Σ (assuming no correlation between features) by category, with values stabilised by adding ϵ.
- To prevent division by zero when computing the variance, a small constant ϵ (epsilon) is added.
- Compute the class posterior p_k for each of the classes.
- Compute the log-likelihood using μ_k *and* Σ from A1.

$$p_k(\mathbf{x}|k) = p_k(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}_k|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)\boldsymbol{\Sigma}_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)^{\mathrm{T}}\right)$$

$$\log p(x|k) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x - \mu_k)^{\mathrm{T}}\Sigma_k^{-1}(x - \mu_k) - \frac{d}{2}\log(2\pi)$$

Assignment 1 & 2

- Define the Discriminant Function:

$$\begin{aligned}\delta_k(\mathbf{x}^*) &= \ln(p(k|\mathbf{x}^*)) = \ln(p_k(\mathbf{x}^*|k)) + \ln(p(k)) - \ln \sum_{l \in C} p_l(\mathbf{x}^*|l) \\ &= -\frac{1}{2}\ln(|\boldsymbol{\Sigma}_k|) - \frac{1}{2}(\mathbf{x}^* - \boldsymbol{\mu}_k)\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}^* - \boldsymbol{\mu}_k)^{\mathrm{T}} + \ln(p(k)) + C\end{aligned}$$

- During prediction, we simply choose the class with the highest $g_k(x)$ value.

$$\hat{k} = \mathit{argmax}\ g_k(x)$$

Assignment 1 & 2

- ML-estimates for the data generated with 95% confidence:

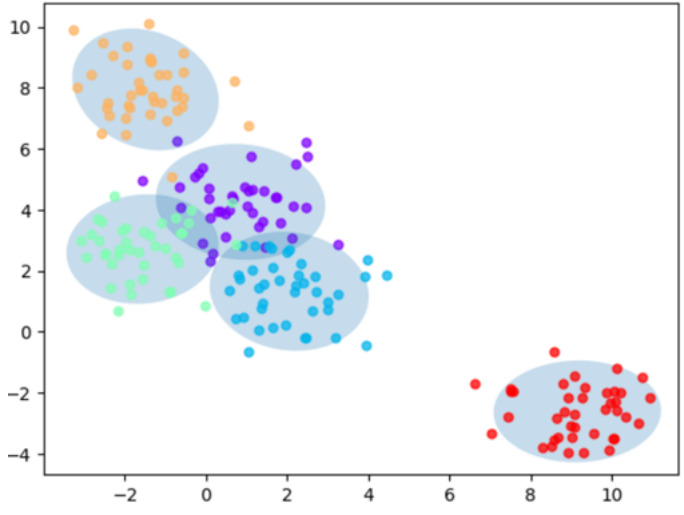

Figure: ML-estimates

Assignment 3 - Encapsulation and Evaluation

- Encapsulate A1 and A2 into a reusable BayesClassifier(train and classify)
- evaluate it multiple times on hierarchically partitioned data, report the test set accuracy (mean $\pm$ standard deviation), and plot the 2D decision boundary.

```
=== Testing Bayes Classifier on Iris ===
Trial: 0 Accuracy 84.4
Trial: 10 Accuracy 95.6
Trial: 20 Accuracy 93.3
Trial: 30 Accuracy 86.7
Trial: 40 Accuracy 88.9
Trial: 50 Accuracy 91.1
Trial: 60 Accuracy 86.7
Trial: 70 Accuracy 91.1
Trial: 80 Accuracy 86.7
Trial: 90 Accuracy 91.1
Final mean classification accuracy  89 with standard deviation 4.16

=== Testing Bayes Classifier on Vowel ===
Trial: 0 Accuracy 61
Trial: 10 Accuracy 66.2
Trial: 20 Accuracy 74
Trial: 30 Accuracy 66.9
Trial: 40 Accuracy 59.7
Trial: 50 Accuracy 64.3
Trial: 60 Accuracy 66.9
Trial: 70 Accuracy 63.6
Trial: 80 Accuracy 62.3
Trial: 90 Accuracy 70.8
Final mean classification accuracy  64.7 with standard deviation 4.03
```

Figure: Accuracy

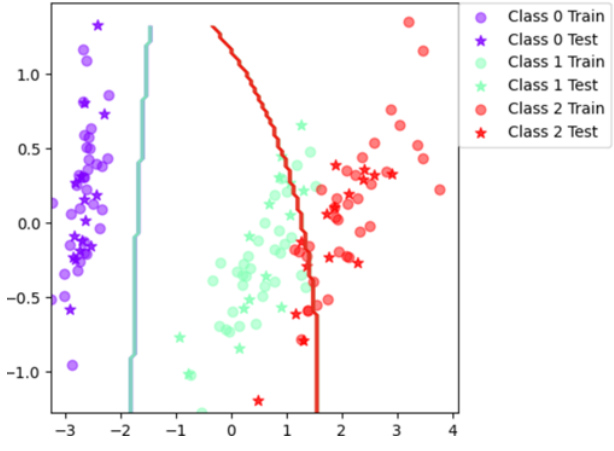

Figure: 2D decision boundary

A3 Q&A

1. When can a feature independence assumption be reasonable and when not?

The assumption of feature independence is valid when features within the data are uncorrelated.

This assumption becomes invalid when features exhibit strong correlation.

2. How does the decision boundary look for the Iris dataset? How could one improve the classification results for this scenario by changing classifier or, alternatively, manipulating the data?

Shape: The boundary of NB is typically a smooth quadratic curve (derived from the quadratic form of Gaussian discrimination).

Improvements:

Adopt more flexible models (decision trees / models post-Boosting);

Perform feature engineering (standardisation/non-linear transformations/PCA);

Tune ensemble models to enhance generalisation.

Assignment 4 - Weighted NB

Modify Assignment1 and Assignment2 to a weighted version incorporating sample weights W.

- In this step, we extend the Naive Bayes classifier so that each training sample has an importance weight.
- Samples that are misclassified will receive higher weights in the next round of training.
- This allows the model to focus more on difficult samples.

Assignment 5 - AdaBoost Bayes Classifier

- Implement and run AdaBoost with weighted Naive Bayes as the base classifier.
- calculating the error rate ϵ_t and classifier weight α_t at each iteration.
- Iteratively update sample weights and perform weighted voting to form the final ensemble model.
- verifying accuracy improvements over the baseline on Iris/Vowel datasets (more pronounced gains observed in complex multi-class tasks).

Assignment 5 - AdaBoost Bayes Classifier

```
=== Testing Boosted Bayes Classifier (T=10) on Iris ===
Trial: 0 Accuracy 95.6
Trial: 10 Accuracy 100
Trial: 20 Accuracy 93.3
Trial: 30 Accuracy 91.1
Trial: 40 Accuracy 97.8
Trial: 50 Accuracy 93.3
Trial: 60 Accuracy 93.3
Trial: 70 Accuracy 97.8
Trial: 80 Accuracy 95.6
Trial: 90 Accuracy 93.3
Final mean classification accuracy  94.7 with standard deviation 2.84

=== Testing Boosted Bayes Classifier (T=10) on Vowel ===
Trial: 0 Accuracy 76.6
Trial: 10 Accuracy 85.7
Trial: 20 Accuracy 83.1
Trial: 30 Accuracy 79.9
Trial: 40 Accuracy 72.7
Trial: 50 Accuracy 76
Trial: 60 Accuracy 81.8
Trial: 70 Accuracy 82.5
Trial: 80 Accuracy 79.9
Trial: 90 Accuracy 83.1
Final mean classification accuracy  80.2 with standard deviation 3.42
```

Figure: Accuracy

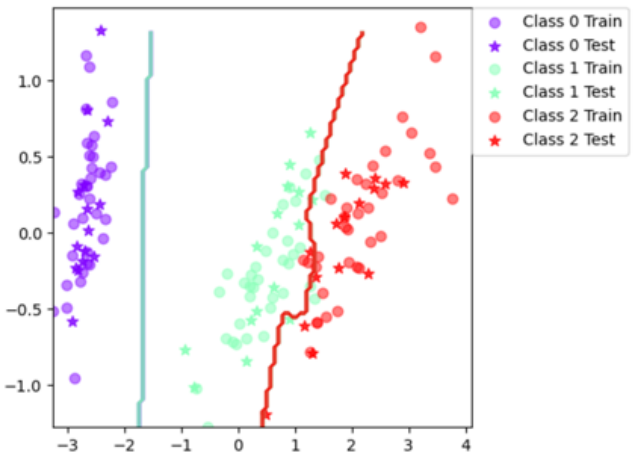

Figure: 2D decision boundary

A5 Q&A

1. Is there any improvement in classification accuracy? Why/why not?

Typically, boosting amplifies the advantages of weak learners incrementally, focusing on challenging examples through sample weighting. This approach reduces training error and often improves test performance. However, if noise or outliers are prevalent, or if the independence assumption of the base learners is severely violated, the gains may be limited or even lead to overfitting.

2. Plot the decision boundary of the boosted classifier on iris and compare it with that of the basic. What differences do you notice? Is the boundary of the boosted version more complex?

More complex. Focusing on marginal samples introduces finer bends and local adjustments to the boundary. Compared to the ‘smooth quadratic boundary’ of basic NB, the boosted version exhibits greater tortuosity.

3. Can we make up for not using a more advanced model in the basic classifier (e.g. independent features) by using boosting?

Partly. Boosting can reduce bias by combining multiple weak learners, but it can’t correct the wrong independence assumption. If features are highly correlated, a stronger base model like a decision tree works better.

Assignment 6 - Decision Tree System Comparison

Change the base learner to a decision tree, compare it with the boosted version, and plot their 2D decision boundaries.

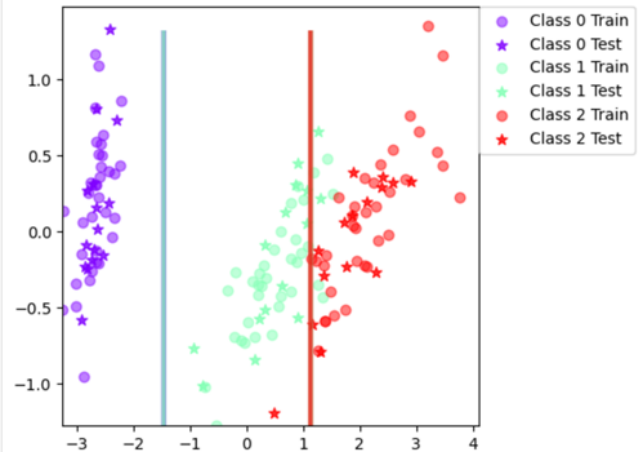

Figure: Single Decision Tree Classifier

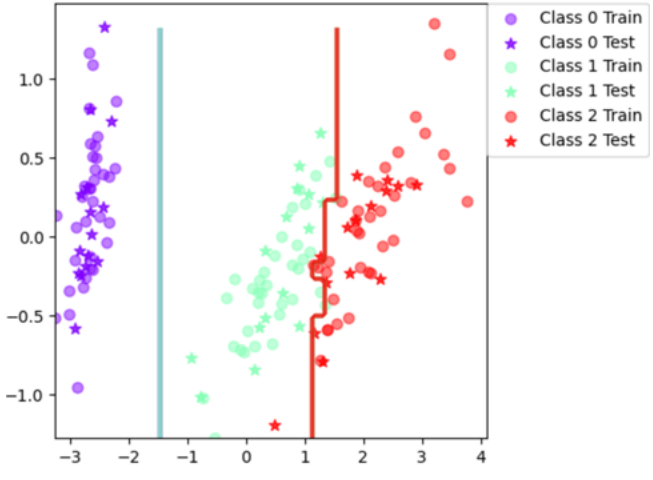

Figure: Boost Decision Tree Classifier

Assignment 6 - Decision Tree System Comparison

To evaluate how boosting improves the performance of a decision tree model by comparing a single decision tree with its boosted version on two datasets (Iris and Vowel).

```
=== Testing Decision Tree Classifier on Iris ===
Trial: 0 Accuracy 95.6
Trial: 10 Accuracy 100
Trial: 20 Accuracy 91.1
Trial: 30 Accuracy 91.1
Trial: 40 Accuracy 93.3
Trial: 50 Accuracy 91.1
Trial: 60 Accuracy 88.9
Trial: 70 Accuracy 88.9
Trial: 80 Accuracy 93.3
Trial: 90 Accuracy 88.9
Final mean classification accuracy  92.4 with standard deviation 3.71

=== Testing Decision Tree Classifier on Vowel ===
Trial: 0 Accuracy 63.6
Trial: 10 Accuracy 68.8
Trial: 20 Accuracy 63.6
Trial: 30 Accuracy 66.9
Trial: 40 Accuracy 59.7
Trial: 50 Accuracy 63
Trial: 60 Accuracy 59.7
Trial: 70 Accuracy 68.8
Trial: 80 Accuracy 59.7
Trial: 90 Accuracy 68.2
Final mean classification accuracy  64.1 with standard deviation 4
```

Figure: Single Decision Tree Classifier

```
=== Testing Boosted Decision Tree (T=10) on Iris ===
Trial: 0 Accuracy 95.6
Trial: 10 Accuracy 100
Trial: 20 Accuracy 95.6
Trial: 30 Accuracy 93.3
Trial: 40 Accuracy 93.3
Trial: 50 Accuracy 95.6
Trial: 60 Accuracy 88.9
Trial: 70 Accuracy 93.3
Trial: 80 Accuracy 93.3
Trial: 90 Accuracy 93.3
Final mean classification accuracy  94.6 with standard deviation 3.65

=== Testing Boosted Decision Tree (T=10) on Vowel ===
Trial: 0 Accuracy 85.7
Trial: 10 Accuracy 90.9
Trial: 20 Accuracy 87
Trial: 30 Accuracy 90.9
Trial: 40 Accuracy 85.1
Trial: 50 Accuracy 79.9
Trial: 60 Accuracy 91.6
Trial: 70 Accuracy 85.7
Trial: 80 Accuracy 88.3
Trial: 90 Accuracy 85.1
Final mean classification accuracy  86.5 with standard deviation 3.02
```

Figure: Boost Decision Tree Classifier

A6 Q&A

Answer questions 1-3 in assignment 5 for the decision tree.

A1: classification accuracy

There is an improvement. Boosting effectively reduces the variance and bias of the individual Decision Tree, leading to better generalization and accuracy.

A2: decision boundary

It is more complex and irregular. It combines multiple simple splits to create non-axis-parallel boundaries, which fit the data more tightly. But beware of overfitting.

A3: Compensate for model deficiencies

Boosting enhances the stability (robustness) of the Decision Tree, preventing a single tree from being overly sensitive to small changes in the training data.However, regularisation techniques such as shallow trees, minimum sample size, and learning rate must be employed to suppress overfitting.

A7 Comprehensive criteria

If you had to pick a classifier, naive Bayes or a decision tree or the boosted versions of these, which one would you pick?

Evaluation Standard	Best/Most Robust Classifier(s)	Core Reason (Concise)
Outliers	**Decision Tree**	Both are relatively robust. **Boosting** is the most sensitive as it gives disproportionate weight to misclassified outliers.
Irrelevant Inputs	**Decision Tree / Boosted DT**	Can **automatically select** the most important features and ignore the irrelevant ones during the splitting process.
Predictive Power	**Boosted Decision Tree**	Possesses the highest capability, combining the Decision Tree's ability to handle **non-linearity** with Boosting's ability to **reduce bias**.
Mixed Types of Data	**Naive Bayes / Decision Tree**	Both are well-suited. Naive Bayes can use different distribution models; Decision Trees inherently handle both continuous and categorical data.
Scalability	**Naive Bayes**	Most **computationally efficient** for large datasets (N large), as training involves a single, quick calculation of means and covariances.